

# Prediktivno vodenje nestabilnega sistema s sprotno identifikacijo verjetnostnega modela

Martin STEPANČIČ, Juš KOCIJAN

**Izvleček:** Predstavljamo metodo prediktivnega vodenja, ki uporablja verjetnostni model in s katerim lahko vodimo tudi nestabilni sistem. Pomen prediktivnega vodenja je sprotno iskanje ustreznega vhodnega signala pri izbrani kriterijski funkciji. Ta funkcija odraža kakovost zaprtozančnega sistema in je odvisna od izbranega vhodnega signala procesa, referenčnega signala in večkoračne napovedi izhodnega signala procesa z njegovim modelom. Za modeliranje dinamike procesa smo uporabili model na podlagi Gaussovih procesov, ki je verjetnostni model črne škatle in izhaja s področja jedrnih metod. Njegova prednost je napovedovanje vrednosti z mero negotovosti in modeliranje brez podrobnejšega poznavanja fizikalnega ali kemijskega ozadja procesa. Negotovost napovedi je izražena z varianco porazdelitve možnih napovedi. Varianco pri vodenju upoštevamo tako, da se med delovanjem izognemo neznanemu območju, ki ga algoritem prepozna po povečani varianci. Po drugi strani pa model identificiramo sprotno in s tem omogočimo prilagoditev modela za območje, ki ni dovolj znano.

**Ključne besede:** dinamični sistemi, avtomatsko vodenje, prediktivno vodenje, model na podlagi Gaussovih procesov, sprotna identifikacija

## 1 Uvod

Področje vodenja sistemov se nenehno razvija glede na obseg tehničnih in ekonomskih problemov v različnih panogah. Načrtovanje vodenja dinamičnih sistemov temelji na poznavanju procesa, ki ga želimo voditi, in sicer v obliki njegovega modela. Priprava natančnega modela dinamičnega sistema na podlagi fizikalnih ali kemijskih zakonov zahteva čas, napor in znanje, ki se razlikuje od procesa do procesa. Alternativa problemu modeliranja na podlagi fizikalnih ali kemijskih zakonov je metoda modeliranja iz merjenega odziva procesa kar imenujemo *eksperimentalno modeliranje* ali *identifikacija*.

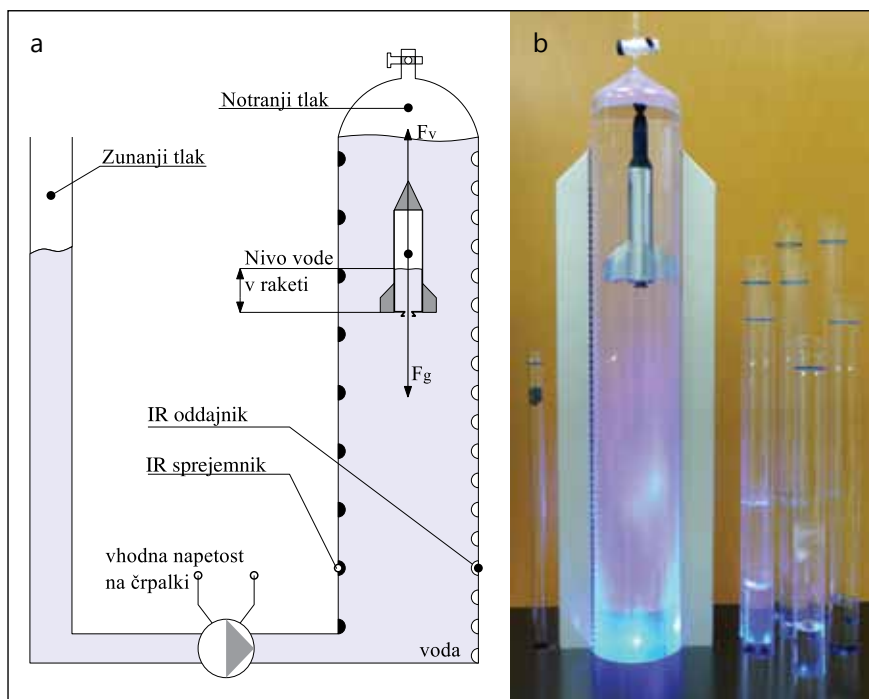
Martin Stepančič, univ. dipl. inž., Institut Jožef Stefan, Mednarodna podiplomska šola Jožefa Stefana, Ljubljana; prof. dr. Juš Kocijan, univ. dipl. inž., Institut Jožef Stefan, Univerza v Novi Gorici

Identifikacija dinamičnih sistemov je zanimiva za načrtovanje raznih metod vodenja. Taka zelo znana metoda vodenja je *prediktivno vodenje*. Od drugih metod se razlikuje po tem, da regulator vsebuje celoten model procesa in ga uporablja za napovedovanje izhodnih signalov procesa v prihodnosti. Z optimizacijo algoritma vodenja poišče najugodnejši vhodni signal v proces, ki ga nato uporabi za naslednji vzorec regulirnega signala.

Naš problem vodenja je naslednji: zasnovati želimo algoritem za vodenje nelinearnega dinamičnega sistema brez vnaprej znanega modela procesa. Cilj je doseči uspešno vodenje izbranega nestabilnega sistema s prediktivnim vodenjem in sprotno identifikacijo modela.

Metodo vodenja bomo prikazali na modelni napravi, ki je prikazana na *sliki 1*. Zgrajena je bila na Inštitutu Jožef Stefan za preizkušanje in demonstriranje metod vodenja [3]. Naprava

je zgrajena iz dveh povezanih shranjevalnikov vode s črpalko med njima. Prvi shranjevalnik je zgoraj odprt valj (ali več valjev) v stiku z zunanjim zračnim tlakom. Drugi shranjevalnik je zgoraj zaprt valj z ujetim zrakom in plovilom v obliki rakete. V raketi, ki ima odprtino na spodnji strani, je zračni mehur. Njena naravna lega je na vrhu valja zaradi prevladujoče sile vzgona. S črpalko vplivamo na spremembo tlaka v zaprtem valju. Volumen raketnega mehura je obratno sorazmeren tlaku ob raketi, zato se sila vzgona zmanjša s povečanjem tlaka ob raketi in obratno. Iz navedenih dejstev lahko zaključimo, da napetostni signal na črpalki vpliva na lego rakete. Višino rakete merimo s senzorjem, ki je sestavljen iz niza infrardečih diod in fototranzistorjev. Senzor višine daje kvantiziran signal, razdeljen na 100 delov, kar bistveno zmanjša natančnost merjenja višine. Modelna naprava je nestabilen in nelinearen proces z enim vhodnim napetostnim signalom na črpalki in izhodnim signalom senzora višine.



Slika 1. Shema modelne naprave (a) in njena fotografija (b)

Razlaga metode vodenja in modeliranja nestabilnega hidravličnega sistema sledi v 2. in 3. poglavju. Izvedba in rezultati vodenja so razloženi v 4. poglavju, nato sledi sklep v 5. poglavju.

## 2 Prediktivno vodenje

### 2.1 Osnovni koncept

Prediktivno vodenje temelji na neposredni rabi modela procesa za

vodenje [1, 8] in določa regulirni signal z optimizacijskim algoritmom, ki minimizira izbrano kriterijsko funkcijo (slika 2).

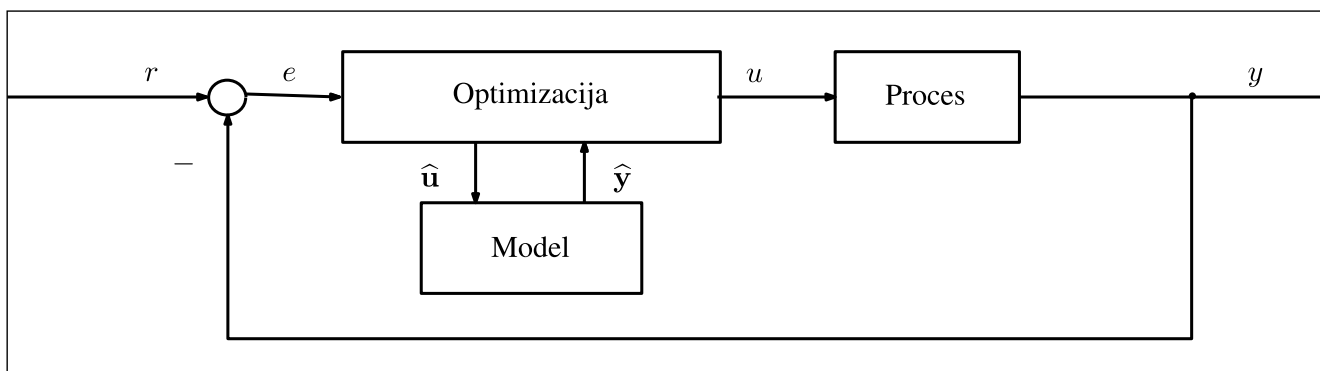
Kriterijska funkcija navadno vključuje napoved izhoda modela procesa  $\hat{y}$  pri nekem vhodnem signalu  $\hat{u}$ . Število časovnih korakov napovedi imenujemo *prediktivni horizont* (slika 3).

Regulirni signal  $\hat{u}$  je vektor vrednosti, ki ga optimiramo glede na vrednost kriterijske funkcije. Številu zaporednih vrednosti regulacijskega signala, ki jih optimiziramo, pravimo *regulirni horizont* in je navadno krajši od prediktivnega horizonta.

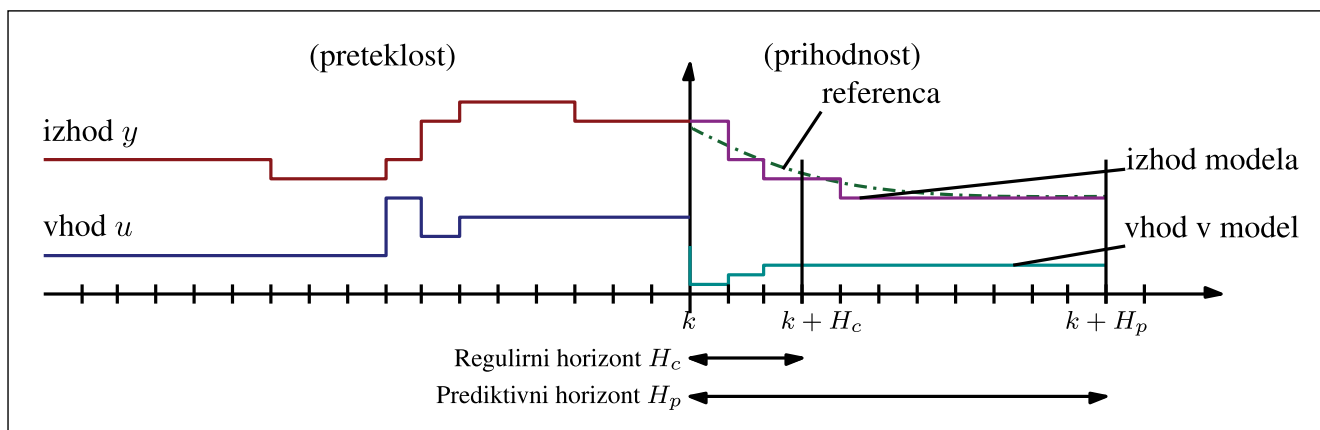
V našem primeru kriterijsko funkcijo opisuje enačba [11]:

$$J = \sum_{j=1}^{H_p} e^2(k+j) + \sigma^2(k+j) \quad (1)$$

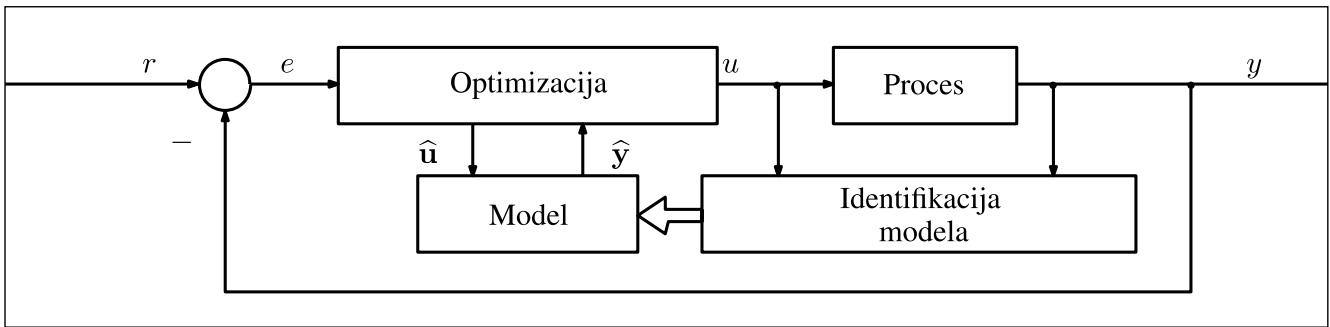
kjer je  $e^2(k+j)$  kvadrat pogreška med napovedjo modela in želenim izhodom procesa v koraku  $(k+j)$  in je  $\sigma^2(k+j)$  varianca napovedi modela v koraku  $(k+j)$ .



Slika 2. Bločna shema prediktivnega vodenja



Slika 3. Prikaz horizontov prediktivnega vodenja, kjer  $k$  predstavlja diskretni časovni korak



Slika 4. Bločna shema adaptivnega prediktivnega vodenja

## 2.2 Adaptivno prediktivno vodenje

Glavna lastnost adaptivnega vodenja je prilagajanje spremenljivim razmeram, pogosto s spreminjanjem parametrov regulatorja [10]. Prediktivno vodenje je adaptivno takrat, ko model procesa identificiramo sproti med delovanjem (slika 4). Adaptivno prediktivno vodenje je uporabno, ko želimo model procesa sproti prilagoditi področju delovanja obravnavanega procesa.

## 3 Sprotno modeliranje procesa

Identifikacija modela je postopek, pri katerem model gradimo na podlagi meritev signala vhoda in izhoda procesa [4]. Identifikacijo

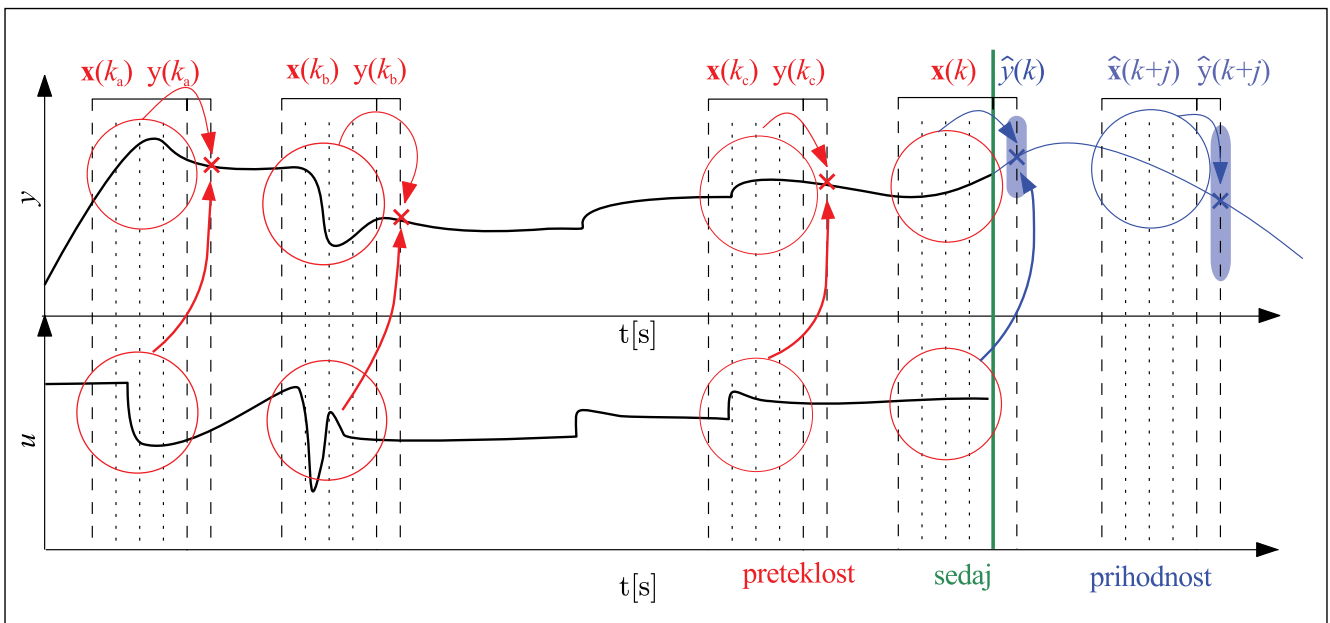
modela lahko izvajamo sproti, da model prilagodimo novim merilom signalov procesa.

Pogoste metode modeliranja nelinearnega dinamičnega sistema so metode, s katerimi dobimo nelinearne modele s posplošenim pogreškom (angl. *Nonlinear autoregressive model with exogenous input* – NARX). Z modelom NARX napovemo vrednost izhoda sistema v odvisnosti od zakasnjenih vrednosti signala vhoda in izhoda sistema. Zakasnjene vrednosti oblikujemo v t. i. regresorski vektor, ki ponazarja stanje dinamičnega sistema. Večkoračno napoved modela izračunamo tako, da trenutno napoved modela obravnavamo kot izmerjeno vrednost in sestavimo regresorski vektor za naslednji časovni korak. Regresorski vektor uporabimo za

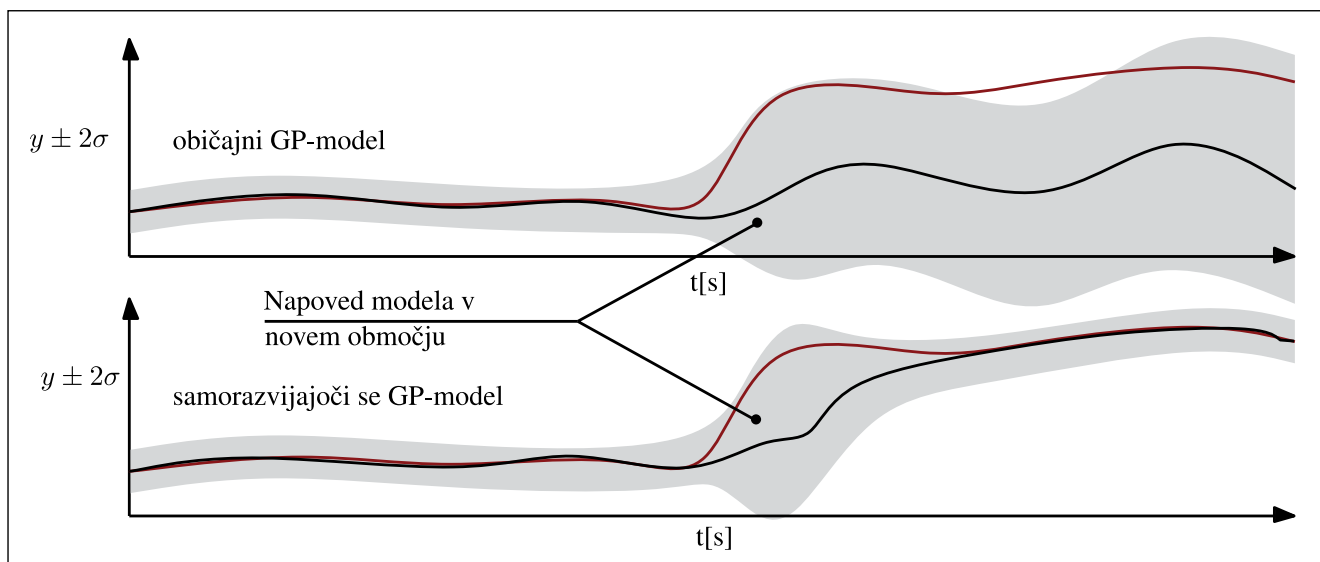
izračun naslednje napovedi in ta postopek ponavljamo za poljubno število časovnih korakov vnaprej (slika 5).

Model NARX lahko zgradimo na različne načine. Ena izmed možnih metod je modeliranje na podlagi Gaussovih procesov (GP), s katero dobimo GP-model. Ta verjetnostna in jedrna metoda modelira sistem neposredno iz vhodnih in izhodnih podatkov. Napoved GP-modela je za razliko od modelov, dobljenih z drugimi metodami, opisana z Gaussovo verjetnostno porazdelitvijo. Več o GP-modelih je razloženo v [1, 2, 9, 10].

Gaussova verjetnostna porazdelitev je določena s srednjo vrednostjo in varianco. Varianco napovedi GP-modela lahko tolmačimo kot



Slika 5. Prikaz večkoračne napovedi z modelom NARX. Označeni krogi predstavljajo regresorske vektorje  $\mathbf{x}$  zakasnjenih vrednosti signala vhoda  $u$  in izhoda  $y$  ob raznih časovnih korakih  $k_a, k_b, k_c, k$  in  $k + j$ . Pripadajoče napovedi regresorskega vektorja  $\hat{\mathbf{y}}$  so označene z modro puščico, standardni odklon verjetnostne porazdelitve napovedi je označen s svetlomodrim pasom.



**Slika 6.** Simbolični prikaz razlike med napovedima časovno nespremenljivega (zgoraj) in časovno spremenljivega GP-modela (spodaj). Slika predstavlja: izhod procesa (rdeča krivulja), srednjo vrednost napovedi (črna krivulja) in dvakratni standardni odklon napovedi (siv pas).

kvantitativno oceno negotovosti modela, na katero vpliva šum uporabljenih meritev ali pomanjkanje meritev izhoda sistema. To je uporabna informacija, ko želimo vedeti, na primer, ali primanjkuje meritev za učinkovito modeliranje.

GP-model lahko nadgradimo tako, da identifikacijo modela izvajamo sproti iz trenutnih meritev signala procesa. GP-model, ki se spreminja z novimi podatki, imenujemo samorazvijajoči se GP-model [5, 6, 7]. Razlika med napovedmi navadnega in samorazvijajočega se GP-modela je prikazana na *sliki 6*. Samorazvijajoči se GP-model se sproti prilagaja dinamiki procesa v trenutnem območju delovanja procesa. Posledica uporabe samorazvijajočega se GP-modela je manjša negotovost napovedi po prehodu v območje, kjer modela še nismo učili (*slika 6* – spodaj).

V praksi je treba omejiti število uporabljenih meritev za identifikacijo modela, da ohranimo manjšo računsko zahtevnost novo identificiranega modela. Zato želimo izbrati le tiste meritve, ki so bogate z informacijami. Metode za optimalni izbor takih meritev so podrobneje razložene v [6, 7].

Prehod iz navadnega, nesprotnega GP-modela, na samorazvijajoči se

GP-model spremeni navadni prediktivni regulator v adaptivnega. Adaptivni prediktivni regulator s samorazvijajočim se GP-modelom smo uporabili za izvedbo vodenja nestabilnega hidravličnega sistema.

#### ■ 4 Primer vodenja nestabilnega dinamičnega sistema

Regulacijo sistema smo izvedli z osebnim računalnikom z operacijskim sistemom Linux. Vhod in izhod naprave smo povezali z računalnikom tako, da smo uporabili vhodno-izhodno enoto za zapis in zajem podatkov, ki bo podrobneje opisana v podpoglavju 4.1. Metoda vodenja in metodo modeliranja smo izvedli s programsko kodo, ki bo razložena v podpoglavju 4.2.

##### 4.1 Vhodno-izhodna enota

Za posredovanje in zajem podatkov smo uporabili modul NI USB-6009 [12] proizvajalca National Instruments, ki ima več analognih in digitalnih kanalov (*slika 7*). Od teh smo uporabili le en analogni vhod za merjenje višine rakete in en analogni izhod za vodenje črpalke. Največja hitrost zajema podatkov je 48 000 vzorcev na sekundo, največja hitrost zapisa na analogni iz-

hod je 150 vzorcev na sekundo. Čas vzorčenja procesa smo izbrali glede na ocenjeno dinamiko procesa 0,1 sekunde.

Gonilnik modula NI USB-6009 je zasnovan kot programski vmesnik, ki ponuja načrtovalcu vgradnjo knjižnic proizvajalca National Instruments v lastno programsko opremo. Slabost gonilnika je nezadostna podpora za operacijski sistem Linux in relativno velika poraba prostora zunanjega pomnilnika. Zato smo razvili lastni gonilnik. Napisali smo ga v programskem jeziku C in uporabili knjižnico *Libusb-1.0* [13]. Postopek priprave lastnega gonilnika je temeljil na imitaciji originalnega. Njegovo de-



**Slika 7.** Vhodno-izhodni modul NI USB-6009

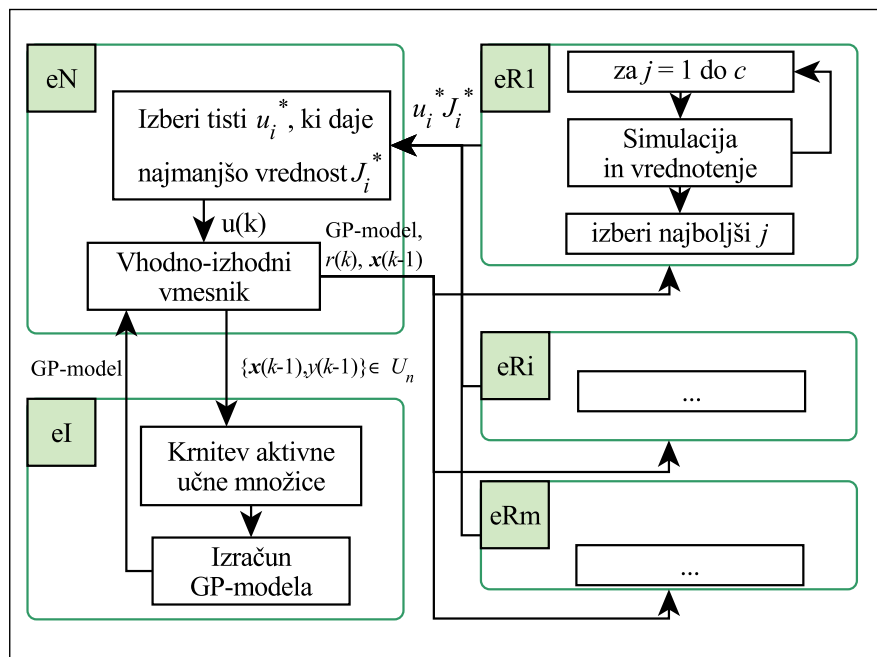
**Preglednica 1.** Pseudokoda porazdeljenega algoritma na ločene enote

Enota eN	Enota eI	Enota eRi, i = 1...m
<p>Neprestano ponavljaj:</p> <ol style="list-style-type: none"> <li>1. Pridobi regresorski vektor <math>x(k-1)</math>,</li> <li>2. primerjaj napoved modela <math>y_n</math> in dejansko vrednost <math>y_d</math></li> <li>3. Če <math> y_n(k-1) - y_d(k-1)  &gt; E</math> potem:             <ol style="list-style-type: none"> <li>a. pošlji regresorski vektor in vrednost <math>y_d</math> enoti eI.</li> </ol> </li> <li>4. Če je proces poslal nazaj novi GP-model:             <ol style="list-style-type: none"> <li>a. sprejmi novi GP-model, sicer: b. obdrži stari GP-model.</li> </ol> </li> <li>5. Za <math>i = 1</math> do <math>m</math>:             <ol style="list-style-type: none"> <li>a. Pošlji referenčni signal in GP-model enoti eRi</li> </ol> </li> <li>6. Sprejmi vrednosti <math>u_i^*</math> in <math>J_i^*</math> od vseh enot eRi in jih sestavi v vektorja <math>u^* = [u_1^*, \dots, u_m^*]</math> in <math>J^* = [J_1^*, \dots, J_m^*]</math>.             <ol style="list-style-type: none"> <li>a. Izberi tisto vrednost regulirnega signala <math>u(k)</math>, ki pripada najnižji vrednosti vektorja <math>J^*</math>.</li> </ol> </li> <li>7. Sinhroniziraj se na časovni korak <math>k</math>.</li> <li>8. Pridobi novo vrednost izhodnega signala <math>y(k)</math> in</li> <li>9. nastavi vhodni signal na izračunano vrednost <math>u(k)</math>.</li> </ol>	<p>Neprestano ponavljaj:</p> <ol style="list-style-type: none"> <li>1. Pridobi množico novih regresorskih vektorjev <math>U_n</math> (od enote eN).</li> <li>2. Če <math>U_n</math> ni prazna množica:             <ol style="list-style-type: none"> <li>a. dodaj k aktivni učni množici <math>U</math> množico <math>U_n</math>.</li> <li>b. Dokler je velikost(<math>U</math>) večja od <math>M_u</math> regresorskih vektorjev:                 <ol style="list-style-type: none"> <li>i. Odstrani iz množice <math>U</math> najstarejši regresorski vektor.</li> </ol> </li> <li>c. Izračunaj nov GP-model iz množice <math>U</math>,</li> <li>d. pošlji novi GP-model enoti eN.</li> </ol> </li> </ol>	<p>Neprestano ponavljaj:</p> <ol style="list-style-type: none"> <li>1. pridobi vektor referenčnih vrednosti <math>r</math> in GP-model (od?? enoti eN),</li> <li>2. razdeli območje regulacijskega signala na <math>m</math> podintervalov, kjer je vsak dodatno razdeljen na <math>c</math> podintervalov tako, da tvorimo matriko:             <math display="block">U = \begin{bmatrix} u_{11} &amp; \dots &amp; u_{1c} \\ \vdots &amp; \ddots &amp; \vdots \\ u_{m1} &amp; \dots &amp; u_{mc} \end{bmatrix}</math> <p>Iz matrike <math>U</math> izberi <math>i</math>-to vrstico regulirnih vrednosti <math>U_i = [u_{i1}, u_{i2}, \dots, u_{ic}]</math>.</p> </li> <li>3. Za <math>j = 1</math> do <math>c</math>:             <ol style="list-style-type: none"> <li>a. simuliraj proces za <math>H_p</math> korakov z GP-modelom pri vhodnem signalu s konstantno vrednostjo <math>u_{ij}</math></li> <li>b. ovrednoti uspešnost poteka simulacije vodenja s kriterijsko funkcijo <math>J</math> in shrani vrednost funkcije <math>J_j</math>.</li> </ol> </li> <li>4. Izberi tisto vrednost <math>u_i^* = u_{ij}</math> ki daje najnižjo vrednost <math>J_i^* = J_j</math> in pošlji vrednosti <math>u_i</math> in <math>J_i^*</math> enoti eN.</li> </ol>

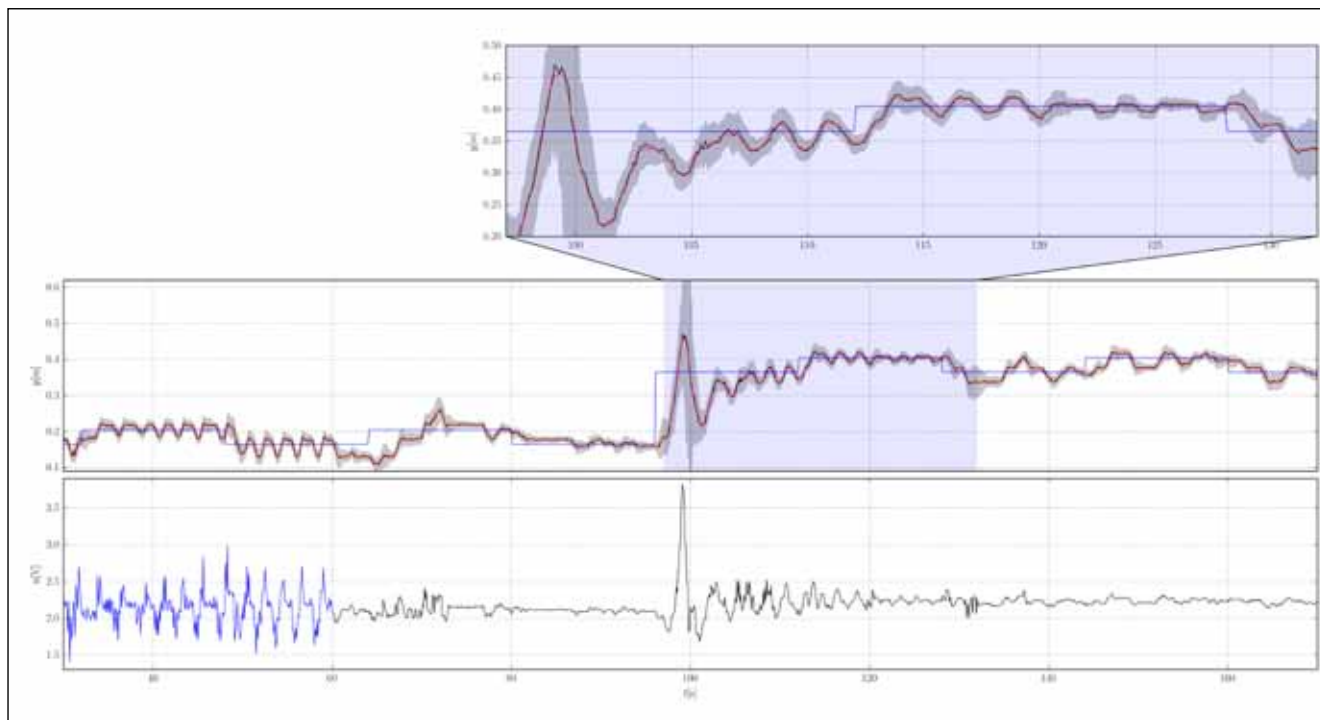
lovanje smo preslikali tako, da smo analizirali komunikacijo na vodilu USB med delovanjem originalnega gonilnika. Izdelani gonilnik smo nato nadgradili za vodenje na daljavo preko omrežnega dostopa.

### 4.2 Izvedba algoritma

Algoritem za vodenje nestabilnega sistema je bil izveden na osebem računalniku z 8-jedrnim procesorjem hitrosti 3,5 GHz in delovnim pomnilnikom velikosti 16 GB z operacijskim sistemom Linux. Ta operacijski sistem je ugodnejši za operacije v realnem času, ker je prioriteta posameznih operacij strogo določena. S tem se izognemo časovnim zakasnitvam pri zajemu in zapisu podatkov na vhodno-izhodni enoti. Koda algoritma je bila izvedena v skriptnem programskem jeziku Matlab®.



**Slika 8.** Paralelizacija algoritma na procese: nadzorna enota (eN) sproti prejema GP-model identifikacijske enote (eI). Enota eN sproži iskanje optimalnega vhodnega signala tako, da pošlje več regulirnim enotam (eRi) trenutni GP-model, trenutno stanje procesa in zeleno vrednost izhoda procesa.



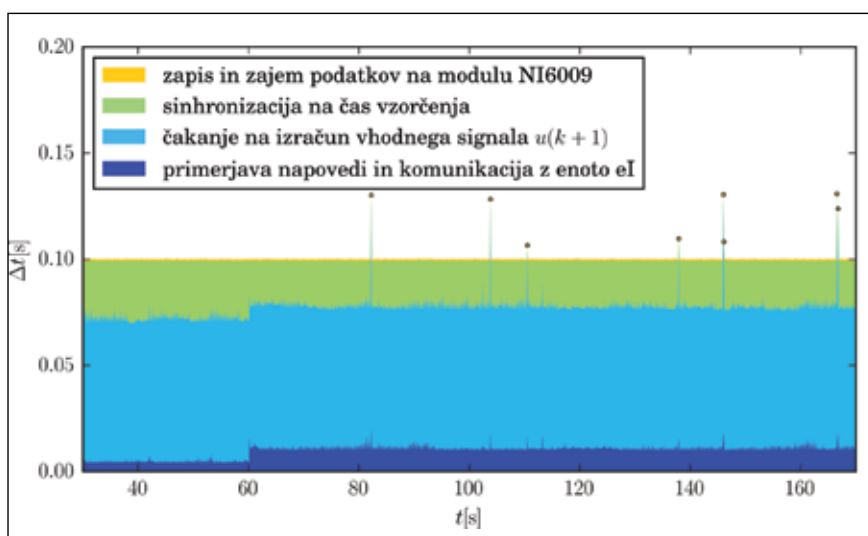
**Slika 9.** Potek prediktivnega vodenja s samorazvijajočim se GP-modelom. Zgornji graf prikazuje višino rakete: izhod procesa (rdeča krivulja), enokoračna napoved modela (črna krivulja) z dvakratnim standardnim odklonom (siv pas okrog črne krivulje), referenca (črtkana modra krivulja). Detajl (zgoraj) zajema potek med 97. in 132. sekundo. Spodnji graf prikazuje regulirni signal PID-regulatorja (modra krivulja), nato regulirni signal prediktivnega vodenja s samorazvijajočim se GP-modelom (črna krivulja).

Algoritem vodenja smo pohitrili tako, da smo ga paralelizirali. Razdelili smo ga po samostojnih enotah: nadzorna enota (eN), identifikacijska enota (eI) in več vzporednih regulacijskih enot (eR<sub>i</sub>), kjer  $i$  pomeni indeks regulacijske enote. Omenjena delitev algoritma na enote omogoča izrazito modularnost, ki omogoča izvedbo sistema

za več procesorskih jeder, vendar taka izvedba zahteva medsebojno komunikacijo in sinhronizacijo. Porazdeljeni algoritem omenjenih enot je prikazana v preglednici 1.

Shema izvedbe algoritma v obliki enot je prikazana na sliki 8. S paralelizacijo izvedbe algoritma smo dosegli zadosti majhen čas izraču-

navanja vhodnega signala, ki mora znašati manj kot 0,1 sekunde. Izračunavanje znotraj časovnega okvira se zahteva le za nadzorno in regulirne enote. Identifikacijski enoti smo dovolili daljši časovni okvir izračunavanja, kar pomeni, da lahko nov GP-model pridobimo šele po nekaj časovnih korakih.



**Slika 10.** Časovna obremenjenost nadzorne enote med vodenjem z delitvijo na komponente časovne obremenjenosti posameznih nalog. Izstopajoče konicе skupne časovne obremenjenosti so označene s piko.

### 4.3 Eksperimentalni rezultati

Potek vodenja nestabilnega hidravličnega sistema je prikazan z regulacijskim signalom  $u$  in izhodnim signalom procesa  $y$  (slika 9). Poleg izhoda procesa je prikazana napoved GP-modela z dvakratnim standardnim odklonom v obliki sivega pasu. Oscilacija signala izhoda procesa na sliki 9 je posledica kvantizacije izhodnega signala. Na sliki 10 je prikazana kumulativna vsota časa izračunavanja v nadzorni enoti eN.

Zaradi varnosti smo sistemu najprej zagotovili stabilno ustaljeno stanje s PID-regulatorjem. V 10. sekundi je pričela delovati sprotna identifikacija modela, nato smo v 60. sekundi

spremenili vodenje tako, da smo zamenjali PID-regulator z adaptivnim prediktivnim regulatorjem.

Iz slike lahko sklepamo, da od 10. do 60. sekunde samorazvijajoči se GP-model pridobi dovolj informacije o sistemu, ker vodenje ostaja vhodno-izhodno stabilno v trenutnem področju delovanja. Zaprtizančni sistem je stabilen tudi po 94. sekundi, ko smo referenčni signal nastavili v območje delovanja, ki ga ne poznamo.

## ■ 5 Zaključek

V prispevku smo predstavili algoritem za adaptivno prediktivno vodenje s samorazvijajočim se GP-modelom. Metodo vodenja smo preizkusili na hidravlični modelni napravi. Iz rezultatov lahko razberemo, da je bila izvedba uspešna kljub brez vnaprej znanega modela za nestabilni, nelinearni proces s kvantiziranim izhodnim signalom.

Pri izvedbi vodenja smo morali biti pozorni na čas izračuna optimalnega regulacijskega signala, ki je moral biti krajši od časa vzorčenja. Računska zahtevnost algoritma za vodenje ni zanemarljiva in kratek čas vzorčenja otežuje izvedbo vodenja sistema. Ta problem smo rešili z izdelavo vzporednih sistemov za hitrejši izračun optimalnega regulacijskega signala.

## Viri

- [1] Juš Kocijan. Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami. Založba Univerze v Novi Gorici, Nova Gorica, 2007.
- [2] Kristjan Ažman. Identifikacija dinamičnih sistemov z Gaussovimi procesi. Doktorska disertacija, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2007.
- [3] Alojzij Pavlinič. Modeliranje in računalniško vodenje nestabilnega laboratorijskega procesa. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko in računalništvo, 1991.
- [4] Drago Matko. Identifikacije. Založba FE in FRI, Ljubljana 1998.
- [5] D. Petelin, J. Kocijan. Control system with evolving Gaussian process models. IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS), 2011, str. 178–184, 2011.
- [6] D. Petelin, J. Kocijan, A. Grancharova. On-line Gaussian process model for the prediction of the ozone concentration in the air. Simulation Modelling Practice and Theory, Vol. 33, str. 68–80, 2013.
- [7] D. Petelin and J. Kocijan. Evolving Gaussian process models for predicting chaotic time-series. IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 8 str., 2014.
- [8] Jan M. Maciejowski. Predictive control with constraints. Pearson Education Limited, Harlow, 2002.
- [9] C. E. Rasmussen, C. K. I. Williams. Gaussian processes for machine learning. MIT Press, Cambridge, MA, London, 2006.
- [10] J. Kocijan. Control Algorithms Based on Gaussian Process Models: A State-of-the-Art Survey, Proceedings Volume from the Special International Conference on Complex Systems: Synergy of Control, Communications and Computing - COSY 2011, str. 69–80 2011.
- [11] Martin Stepančič. Vodenje s samorazvijajočimi modeli na podlagi Gaussovih procesov. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2013.
- [12] User guide and specifications NI-USB6008/6009, National Instruments Corporation, 2005. Dosegljivo: <http://www.ni.com/pdf/manuals/371303m.pdf> [Dostopano 24. 4. 2014].
- [13] Libusb-1.0 API reference. Sourceforge. Dosegljivo: <http://libusb.sourceforge.net/api-1.0/> [Dostopano 11. 7. 2013].

### Model predictive control of an unstable dynamic system with online probabilistic identification

**Abstract:** A model predictive control algorithm with probabilistic model is presented which enables the control of open-loop unstable dynamic systems. The role of this method is to determine a suitable control signal of the process by minimizing a predetermined cost function. The cost function should reflect the performance of the closed-loop system. It depends on the horizon of the input signal, the reference signal, and a multi-step prediction of the system output. The latter's accuracy depends on the quality of the model which in our case is a Gaussian process model that is a probabilistic and black-box model. Its advantage is the prediction with a measurement uncertainty, and the modelling without knowing the physical or chemical background of the process. The prediction uncertainty is given as the variance of the prediction distribution. The estimated variance helps to avoid leading the process into unknown operating regions. On the other hand, the model adapts to a new operating region with an online model identification.

**Keywords:** dynamic system, automatic control, model-predictive control, Gaussian-process model, online identification